MANUAL DE INSTALACIÓN DEL SERVIDOR SIERRA

1. Instalación del servidor de bases de datos geográficas 1
1.1. Instalación y configuración de PostgreSQL 1
1.2. Instalación de Proj y GEOS 2
1.3. Instalación de PostGIS 4
2. Creación de la base de datos 5
3. Instalación y configuración del servidor Apache Geronimo 7
3.1. Instalación y arranque manual 7
3.2. Inicio del servidor con el arranque del sistema 8
3.3. Creación de Database Pools y despliegue de aplicaciones en Geronimo 9
3.4. Gestión de librerías del repositorio de Geronimo 14

Capítulo 1. Instalación del servidor de bases de datos geográficas

1.1. Instalación y configuración de PostgreSQL

La instalación de PostgreSQL 8.1 se puede realizar desde Synaptic, buscando el paquete *postgresql-8.1* y haciendo doble click sobre él y posteriormente click en el botón "Aplicar". Si lo instalamos desde apt-get, ejecutamos la siguiente orden en el terminal:

$ sudo apt-get install postgresql-8.1

Una vez instalado PostgreSQL, pasamos a configurar el usuario `postgres` y a crear la base de datos.

Primero cambiamos el password del usuario `postgres` con la orden:

$ sudo passwd postgres

El prompt nos pide que escribamos el password para el usuario `postgres` dos veces. Para comprobar que la instalación ha sido satisfactoria accedemos a la shell del servidor de bases de datos con la siguiente orden:

$ sudo su postgres -c "psql template1"

Si el acceso ha sido satisfactorio cambiamos la contraseña al usuario predeterminado del servidor de bases de datos con la orden (dentro del cliente PostgreSQL):

template1=# ALTER USER postgres WITH PASSWORD 'nueva_contraseña';

cambiando **nueva_contraseña** por la contraseña que hayamos elegido. Salimos del shell del servidor de la base de datos (orden **\q** del shell del cliente PostgreSQL).

Ahora pasamos a la configuración del servidor. Para permitir conexiones remotas editamos un fichero de configuración de PostgreSQL:

$ sudo gedit /etc/postgresql/8.1/main/postgresql.conf

Buscamos la siguiente línea:

```
#listen_addresses = 'localhost'
```

y la sustituimos por:

```
listen_addresses = '*'
```

También cambiamos la línea:

```
# password_encryption = on
```

por:

```
password_encryption = on
```

Guardamos los cambios editados y reiniciamos el demonio para que los cambios surtan efecto:

$ sudo /etc/init.d/postgresql-8.1 restart

Para configurar la lista de acceso deberemos editar el fichero `/etc/postgresql/8.1/main/pg_hba.conf`. Para permitir a un host, por ejemplo, el acceso, añadiríamos al principio la línea:

```
# TYPE            DATABASE          USER              CIDR-ADDRESS       METHOD

host              all               all               <lista_maquinas>   md5
```

Esto significa que cualquier usuario que acceda desde `<lista_maquinas>` puede acceder a cualquier base de datos utilizando identificación mediante md5 (`<lista_maquinas>` puede ser una lista de ip, o `*` para permitir conexiones desde cualquier máquina).

Para guardar los cambios volvemos a reiniciar el demonio para que los cambios surtan efecto:

$ sudo /etc/init.d/postgresql-8.1 restart

A continuación hay que introducir el directorio con las bibliotecas de ejecución de PostgreSQL en la ruta de búsqueda del enlazador dinámico, para ello hay que introducir la siguiente línea:

```
/usr/lib/postgresql/8.1/lib
```

editando uno de los dos ficheros siguientes (depende cada distribución) como se muestra:

$ sudo gedit /etc/ld.so.conf

ó

$ sudo gedit /etc/ld.so.conf.d/i486-linux-gnu

Luego ejecutamos:

$ sudo ldconfig -v

para actualizar la localización de las librerías.

1.2. Instalación de Proj y GEOS

Para instalar Proj descargamos los fuentes de la dirección:

ftp://ftp.remotesensing.org/proj/proj-4.5.0.tar.gz

Una vez estemos en el directorio donde se ha descargado el fichero, lo descomprimimos con la siguiente orden:

$ tar xvfz proj-4.5.0.tar.gz

Entramos en el nuevo directorio creado (**$ cd proj-4.5.0**) y procedemos a compilar e instalar:

$ sudo ./configure

Si durante el proceso de configuración el sistema no encuentra el compilador o alguna biblioteca de desarrollo, habrá que instalarlas a mano y volver a ejecutar la orden anterior. Para instalar las órdenes que pueden faltar:

$sudo apt-get install gcc make build-essential

Después de la configuración de la compilación procedemos a compilar e instalar:

$ sudo make

$ sudo make install

Siguiendo el mismo proceso explicado en la sección anterior, se añade al final del fichero de rutas del enlazador la siguiente línea para que encuentre las librerías de Proj:

```
/usr/local/lib
```

Ejecutamos la orden:

$ sudo ldconfig -v

Ahora pasamos a instalar GEOS. Bajamos los fuentes de la siguente dirección:

http://geos.refractions.net/geos-3.0.0rc4.tar.bz2

Pasamos a descomprimirlos mediante la siguiente orden, entrando previamente en el directorio donde se ha descargado el fichero:

$ bzcat geos-3.0.0rc4.tar.bz2 | tar xv

Entramos al directorio geos-3.0.0rc4 (**$ cd geos-3.0.0rc4**) y compilamos e instalamos:

$ sudo ./configure

$ sudo make

$ sudo make install

Ejecutamos la siguiente orden para actualizar la localización de las librerías:

$ sudo ldconfig -v

1.3. Instalación de PostGIS

Antes de pasar a compilar los fuentes de PostGIS debemos de instalar algunos paquetes en Ubuntu. Podemos instalarlos buscándolos en Synaptic o con la orden **apt-get**.

- Para Synaptic buscamos los paquetes `build-essential`, `postgresql-server-dev-8.1` y `flex`, y los instalamos haciendo doble click sobre ellos y haciendo click posteriormente en "Aplicar".
- Si lo hacemos desde apt-get ejecutamos la siguiente orden:

  $ sudo apt-get install build-essential postgresql-server-dev-8.1 flex

Una vez realizado el paso anterior pasamos a instalar PostGIS. Descargamos los fuentes de la siguiente dirección:

http://postgis.refractions.net/download/postgis-1.2.1.tar.gz

Nos situamos en el directorio donde se ha descargado el fichero y lo descomprimimos con la siguiente orden:

$ tar xvfz postgis-1.2.1.tar.gz

Entramos dentro del directorio postgis-1.2.1 (**$ cd postgis-1.2.1**) y empezamos a configurar la instalación:

$ sudo ./configure

Cuando termine de configurar la instalación procedemos a compilar:

$ sudo make

Una vez terminado, cambiamos al usuario `postgres` y chequeamos la instalación:

$ sudo su postgres

$ make check

Al final de la ejecución de la orden "make check" nos mostrará un mensaje con los aciertos y los fallos. Si todos los pasos se han realizado correctamente nos mostrará cero fallos. Para instalar PostGIS en el árbol de directorios ejecutaremos:

$ sudo make install

Posteriormente actualizaremos la ruta del enlazador de librerías con la siguiente orden:

$ sudo ldconfig -v

Capítulo 2. Creación de la base de datos

Para crear la base de datos espacial con la información ya introducida simplemente ejecutaremos el script **creabd** como usuario `postgis` (el administrador de PostgreSQL). La sintaxis de esta orden es:

$ ./creabd nombre_base_de_datos directorio_fuentes_postgis

- **nombre_base_de_datos**: es el nombre que queremos que tenga nuestra base de datos.
- **directorio_fuentes_postgis**: es el directorio donde hemos descomprimido los fuentes de PostGIS, y posteriormente hemos compilado e instalado el paquete. Esto es necesario porque el script utiliza ciertos ficheros para crear la base de datos con funcionalidad espacial que se encuentran en este directorio.

Cuando el proceso de creación de la base de datos haya terminado correctamente nos aparecerá el siguiente mensaje:

```
¡ATENCIÓN! Recuerda actualizar el fichero pg_hba.conf para dar permisos al usuario reader.
```

En este paso deberemos dar permisos al usuario reader editando el fichero `/etc/postgresql/8.1/main/pg_hba.conf`. Las reglas para dar permisos se estructuran de la siguiente manera, descrita brevemente en el anterior capítulo:

```
# TYPE             DATABASE          USER              CIDR-ADDRESS       METHOD

tipo_acceso        nombre_bd         usuario           <lista_maquinas>   metodo_aut
```

Donde:

- `tipo_acceso`: denota el lugar desde donde se accede a la base de datos. Los valores son: `local` (acceso a la base de datos desde el ordenador local donde se encuentra instalado PostgreSQL), `host` (acceso desde un host externo), `hostssl` y `hostnossl` (conexiones desde un host externo que utilicen/no utilicen ssl).
- `nombre_bd`: el nombre de la base de datos que es accedida. También se puede poner `all` para denotar acceso a cualquier base de datos.
- `usuario`: el usuario que está accediendo a la base de datos. Este valor también puede ser `all` para denotar acceso de cualquier usuario.
- `<lista_maquinas>`: rango de ip's de acceso para este registro. El formato es `IP/Número` para denotar un rango de IPs, en el que el número indican los bits a 1 por la izquierda de la máscara (por ejemplo, un 8 sería equivalente a una máscara `255.0.0.0`). Para especificar cualquier IP se usa el valor `0.0.0.0/0`. El formato de la IP puede ser también de IPv6. Este campo también se puede dividir en dos columnas: una dirección IP y una máscara en formato IP (por ejemplo `255.255.255.0`).

- `metodo_aut`: el método de autenticación utilizado para acceder. Existen varias posibilidades, pero las que más se utilizan son: `md5` (acceso mediante password cifrado con MD5) y `ident sameuser` (para los accesos desde local, utiliza la contraseña del usuario del sistema que está accediendo).

Por ejemplo, para la base de datos `sierra` y el usuario `reader` (que es el usuario que crea el script automáticamente para acceder a la base de datos), un posible registro de acceso sería:

```
# TYPE            DATABASE          USER              CIDR-ADDRESS       METHOD

host              sierra            reader            0.0.0.0/0          md5
```

Con esta línea le estamos diciendo que si hay un acceso desde cualquier host externo a la base de datos `sierra` con el usuario `reader`, y que el usuario se autentifica con contraseña cifrada, permitir el acceso (siempre que la contraseña proporcionada sea correcta). Es conveniente que las líneas que añadamos se añadan al principio, ya que la primera línea que se encuentra que coincide con los parámetros de acceso es la única que considera. Si todo va bien se permite el acceso, pero si la línea coincide y se deniega el acceso ya no se miran las siguientes líneas del fichero y, por lo tanto, se deniega el acceso definitivamente (aunque en líneas inferiores se permita).

Capítulo 3. Instalación y configuración del servidor Apache Geronimo

3.1. Instalación y arranque manual

Para instalar el servidor Apache Geronimo nos bajamos el paquete con los binarios de la dirección:

http://apache.rediris.es/geronimo/2.0-M5/geronimo-tomcat6-jee5-2.0-M5-bin.tar.gz

Colocamos este fichero en el directorio padre de donde estará instalado Geronimo y procedemos a descomprimir:

$ tar xvfz geronimo-tomcat6-jee5-2.0-M5-bin.tar.gz

Para arrancar Geronimo (suponiendo que tenemos Java correctamente instalado y configurado), nos situamos en el directorio de instalación de Geronimo y dentro de éste en el directorio `bin`, y ejecutamos:

$ ./geronimo.sh start

Ahora podemos acceder vía navegador para configurar el servidor y para desplegar nuestras aplicaciones web. La URL para acceder al servidor debe ser del tipo http://nombre_del_host:8080. Se nos mostraría la pantalla de inicio.

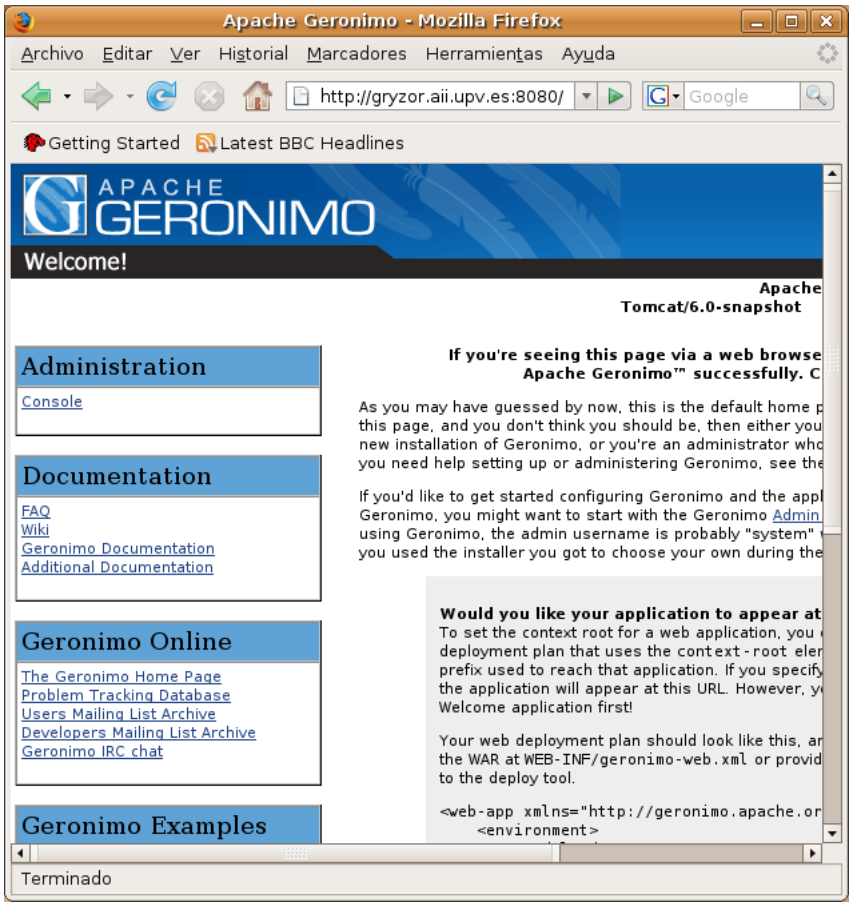
Apache Geronimo - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
http://gryzor.aii.upv.es:8080/
Google
Getting Started
Latest BBC Headlines
APACHE GERONIMO
Welcome!
Apache Tomcat/6.0-snapshot
Administration
Console
Documentation
FAQ
Wiki
Geronimo Documentation
Additional Documentation
Geronimo Online
The Geronimo Home Page
Problem Tracking Database
Users Mailing List Archive
Developers Mailing List Archive
Geronimo IRC chat
Geronimo Examples
If you're seeing this page via a web browse
Apache Geronimo™ successfully. C
As you may have guessed by now, this is the default home p
this page, and you don't think you should be, then either you
new installation of Geronimo, or you're an administrator who
you need help setting up or administering Geronimo, see the
If you'd like to get started configuring Geronimo and the appl
Geronimo, you might want to start with the Geronimo Admin
using Geronimo, the admin username is probably "system"
you used the installer you got to choose your own during the
Would you like your application to appear at
To set the context root for a web application, you
deployment plan that uses the context-root eler
prefix used to reach that application. If you specify
the application will appear at this URL. However, y
Welcome application first!
Your web deployment plan should look like this, ar
the WAR at WEB-INF/geronimo-web.xml or provid
to the deploy tool.
<web-app xmlns="http://geronimo.apache.or
<environment>
Terminado

Figura 3.1. Pantalla de inicio

3.2. Inicio del servidor con el arranque del sistema

Pasamos a configurar el servidor Apache Geronimo para que arranque y para que se desconecte al encender la máquina. Para ello suponemos que Geronimo está instalado en `/usr/local/geronimo2`. Primero copiamos el script `geronimo2.sh` en `/etc/init.d`:

$ sudo cp geronimo2.sh /etc/init.d

Para el correcto funcionamiento debemos editar el script recién instalado. En las líneas donde se exporta la variable `JAVA_HOME` debemos asignar la ruta de donde se encuentra instalado Java en nuestro sistema.

Una vez lo tengamos copiado y correctamente editado, tenemos que crear los enlaces simbólicos para los diferentes niveles de ejecución. Para ello ejecutamos las órdenes:

$ sudo update-rc.d geronimo2.sh defaults

Una vez reiniciemos el sistema, se arrancará el servidor Apache Geronimo. Podemos también ejecutar el script de inicio desde una terminal, dándole las órdenes (como argumentos) **stop**, **start** o **restart**, que paran, inician, o reinician el servicio respectivamente.

3.3. Creación de Database Pools y despliegue de aplicaciones en Geronimo

Para entrar en la consola de administración, en el menú de la izquierda pulsamos sobre el enlace "Console". Esto nos lleva a una pantalla en la que deberemos especificar el usuario y la contraseña del administrador. Los valores que trae por defecto el servidor son: usuario `system`, contraseña `manager`. Una vez hemos accedido al modo de administración podremos cambiar la contraseña si lo deseamos. Para cambiar la contraseña del usuario `system` haremos click en "Console Realms" en el menú de la izquierda. Nos aparecerán dos usuarios, `system` y `admin`. Pulsamos en "Details "en el usuario `system`, y en los dos cuadros que aparecen escribimos y confirmamos la nueva contraseña. Luego pulsamos en Update y, si todo ha ido bien, el cambio de contraseña estará realizado.

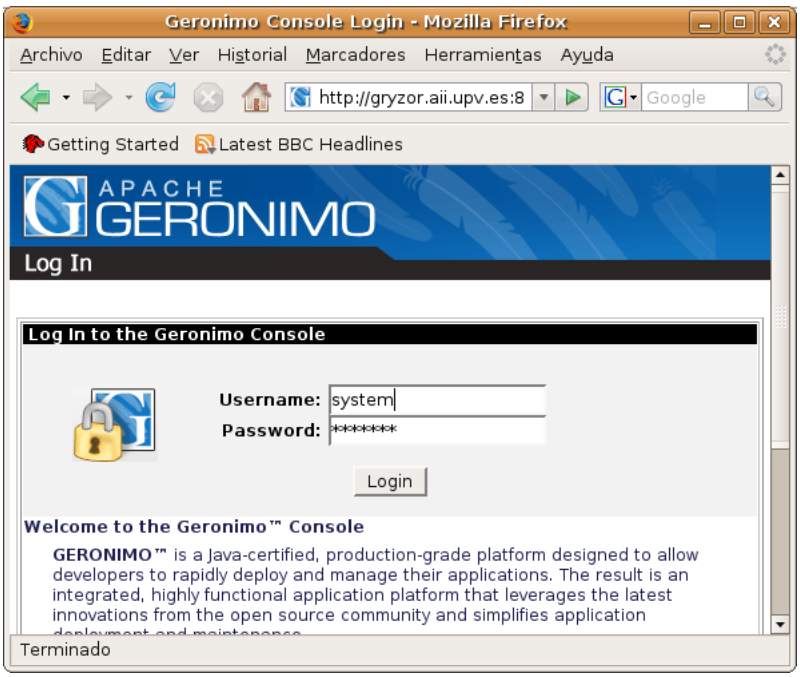
Geronimo Console Login - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
http://gryzor.aii.upv.es:8
Google
Getting Started
Latest BBC Headlines
APACHE GERONIMO
Log In
Log In to the Geronimo Console
Username: system
Password: *******
Login
Welcome to the Geronimo™ Console
GERONIMO™ is a Java-certified, production-grade platform designed to allow developers to rapidly deploy and manage their applications. The result is an integrated, highly functional application platform that leverages the latest innovations from the open source community and simplifies application
Terminado

Figura 3.2. Pantalla de login

Una vez dentro, podemos pasar a configurar distintos aspectos de nuestro servidor. Lo que más nos interesa es crear los Database Pools en Geronimo para que Geronimo maneje la conexión a la base de datos, y nuestras aplicaciones sólo tengan que llamar a Geronimo con el nombre del Database Pool para que éste le devuelva la referencia al objeto conexión.

Para instalar un Database Pool, en el menú de la izquierda hacemos click en "Database Pools". Hacemos click sobre el enlace "Using the Geronimo database pool wizard" para iniciar el asistente de creación del Database Pool.

Figura 3.3. Pantalla de listado y creación de Database Pools

En la casilla "Name of Database Pool" escribimos el nombre que queramos para el Database Pool, y en la lista desplegable de abajo seleccionamos el tipo de base de datos, en nuestro caso PostgreSQL.

Como Geronimo no tiene instalado en su repositorio el driver de PostgreSQL, en la siguiente pantalla hacemos click en "Download a driver". En la lista "Select Driver" seleccionamos PostgreSQL 8.1, esperamos a que el driver esté descargado y volvemos a la pantalla anterior, pero ahora tenemos seleccionado el driver que acabamos de descargar.

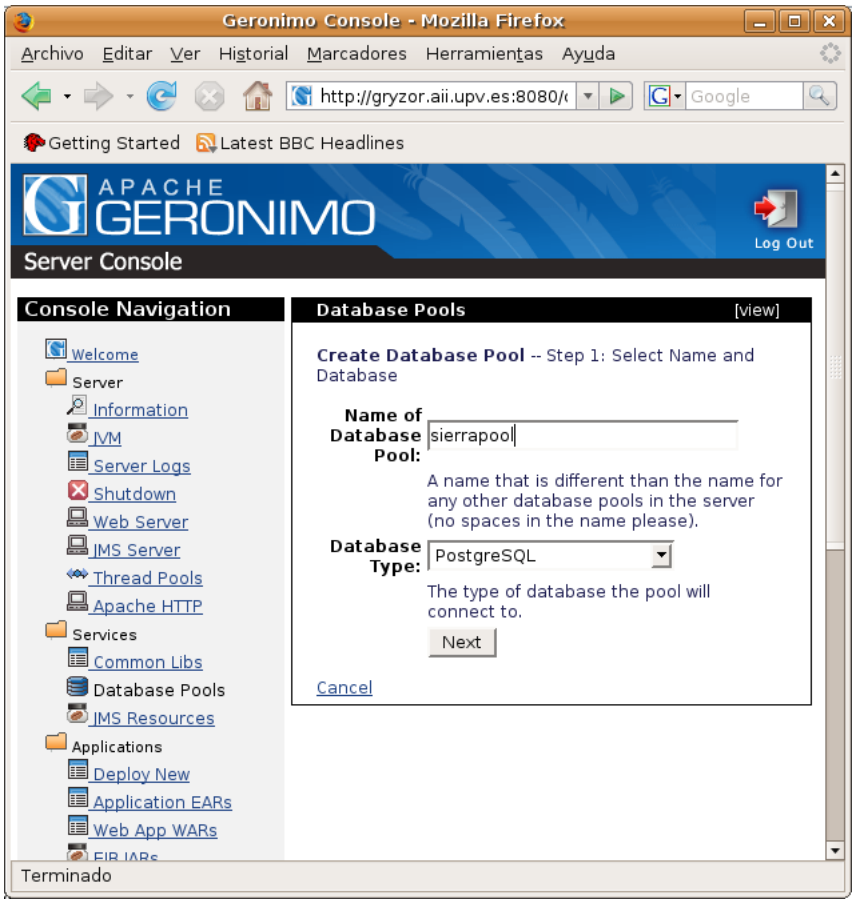

Figura 3.4. Pantalla de selección de driver de base de datos y de nombre del Database Pool

Rellenamos los campos de acceso a nuestra base de datos (nombre de usuario de acceso, contraseña, host, nombre de base de datos y puerto). Cuando hayamos rellenado los campos pulsamos "Next".

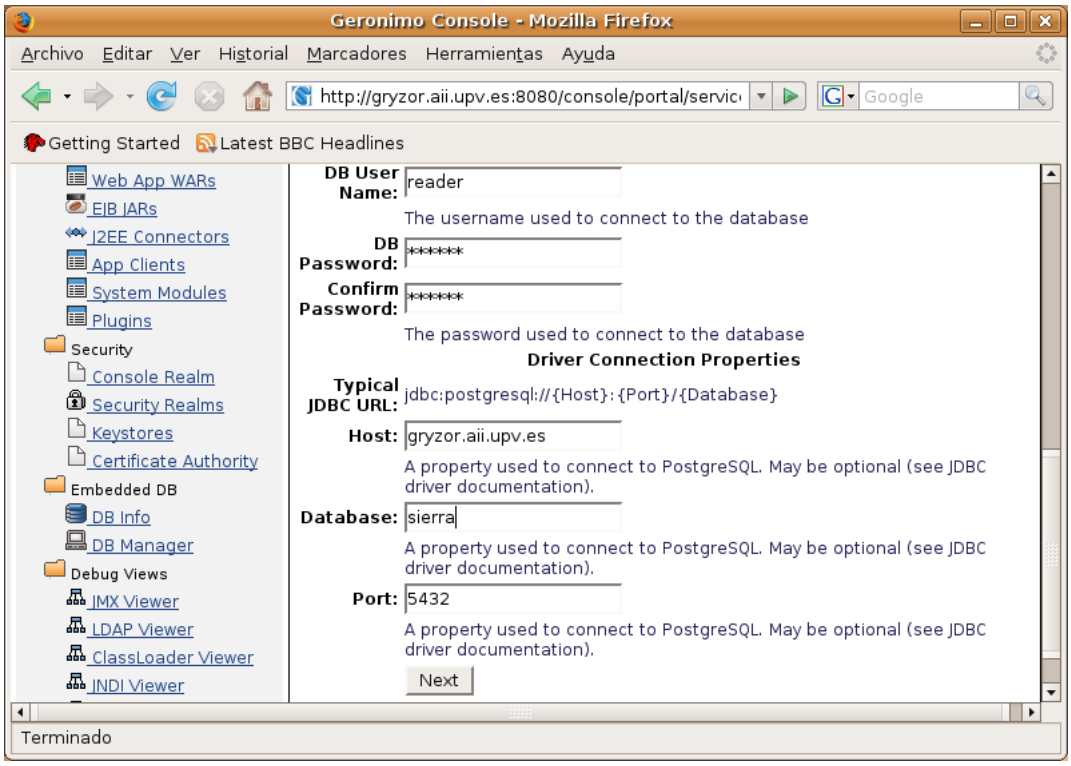

Figura 3.5. Pantalla de introducción de datos sobre la conexión a la base de datos

En la siguiente pantalla hacemos click en "Test connection" para probar que la conexión del Database Pool que está siendo creado funciona correctamente. Si todo ha funcionado correctamente nos aparecerá el mensaje "Test result: Connected to PostgreSQL 8.1.8". En este caso, para finalizar la creación del Database Pool hacemos click en "Deploy". Después nos aparece una lista con todos los Database Pools, debemos comprobar que el nuestro aparece en la lista.

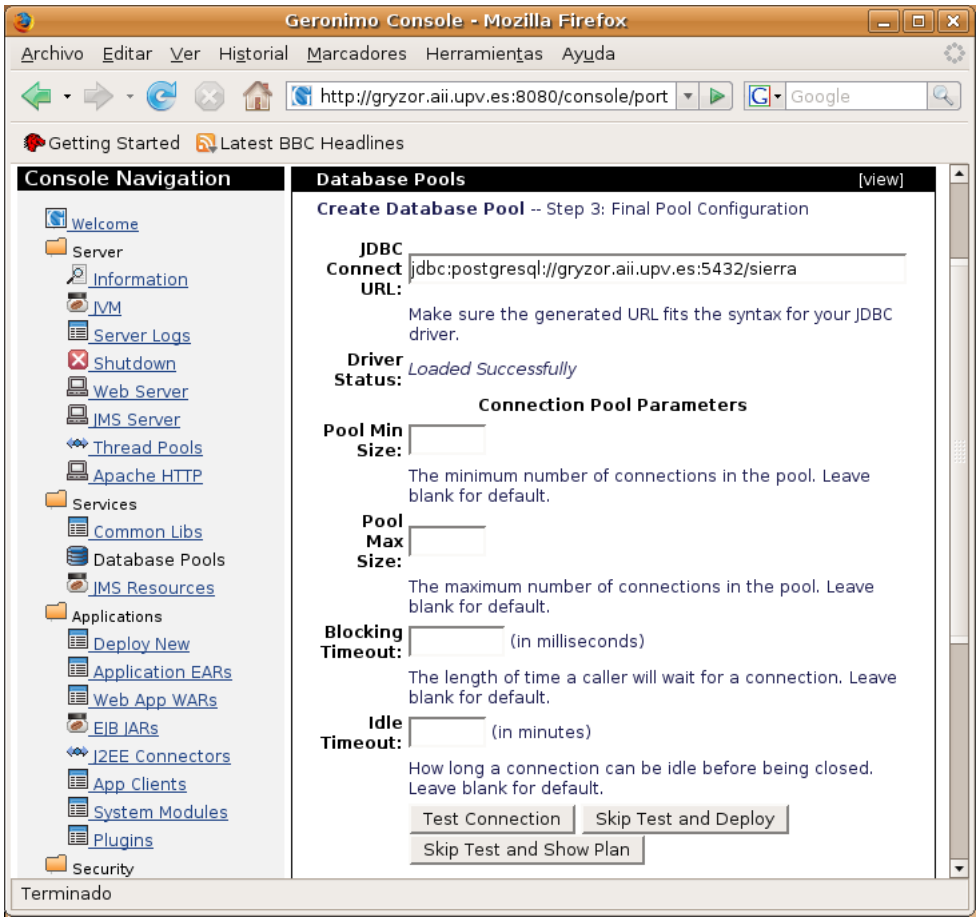

Figura 3.6. Pantalla de configuración del Database Pool

Los datos para crear el Database Pool del ejemplo anterior son:

- Nombre del Database Pool: sierrapool
- Usuario de la base de datos: reader
- Contraseña: <contraseña>
- Host: gryzor.aii.upv.es
- Nombre de la base de datos: sierra
- Puerto: 5432(el que viene por defecto en el asistente)

Para subir nuestras aplicaciones web, en la consola de administración hacemos click en Deploy New. Hacemos click sobre "Examinar", en la casilla nombrada como "Archive", para seleccionar la aplicación web (extensión .war). Comprobamos que está seleccionada la opción "Start app after install" y hacemos click en "Install". Cuando el despliegue de la aplicación web haya terminado nos aparecerán los mensajes "The application was succesfully deployed" y "The application was succesfully started" si todo ha ido correctamente.

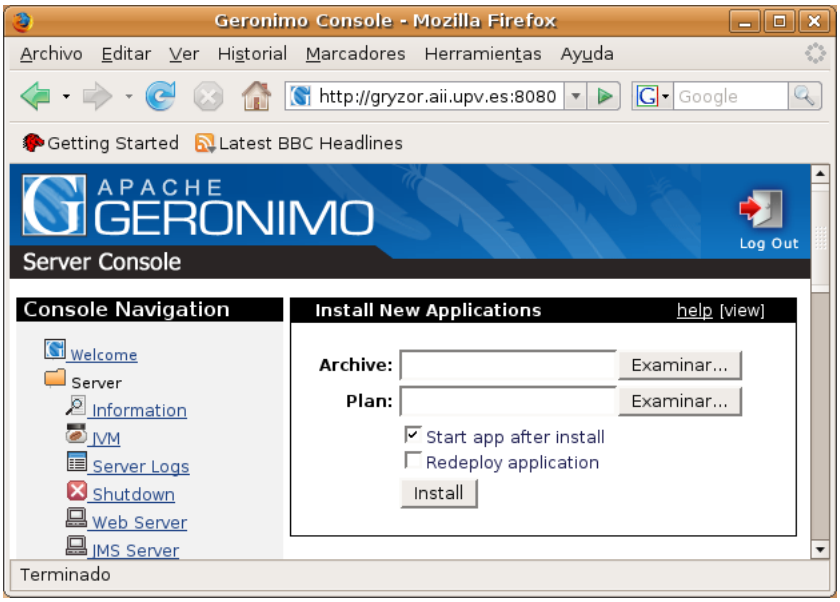

Figura 3.7. Pantalla de despliegue de aplicaciones web

Si queremos redesplegar una aplicación web el proceso sería el mismo que para subir una aplicación web pero marcando esta vez la casilla "Redeploy application".

3.4. Gestión de librerías del repositorio de Geronimo

Para instalar una nueva librería haremos click en "Common Libs", en el menú de la izquierda de la consola de Geronimo. Nos aparecerá un formulario con varios campos. El primero es el campo que indica la ruta de la librería que queremos instalar. Pulsamos en "Examinar" y seleccionamos la librería que queramos añadir a Geronimo.

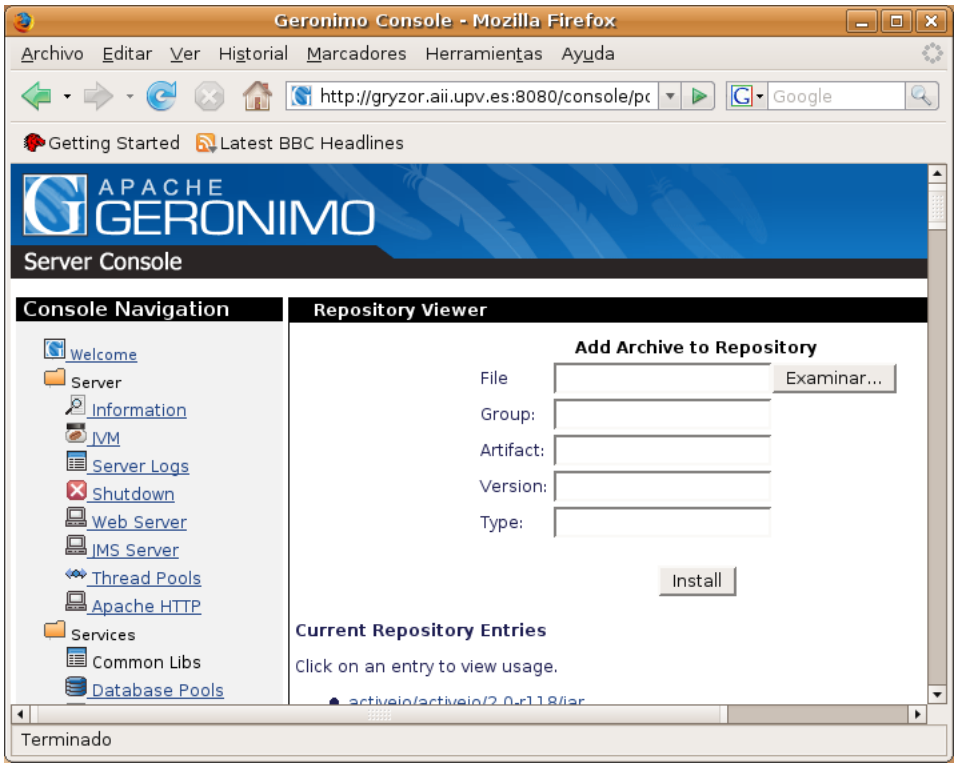

Figura 3.8. Añadiendo librerías

Una vez seleccionada la librería, vemos que nos ha rellenado automáticamente los campos Artifact, Version y Type. En el campo Group podemos poner el nombre genérico de la librería.

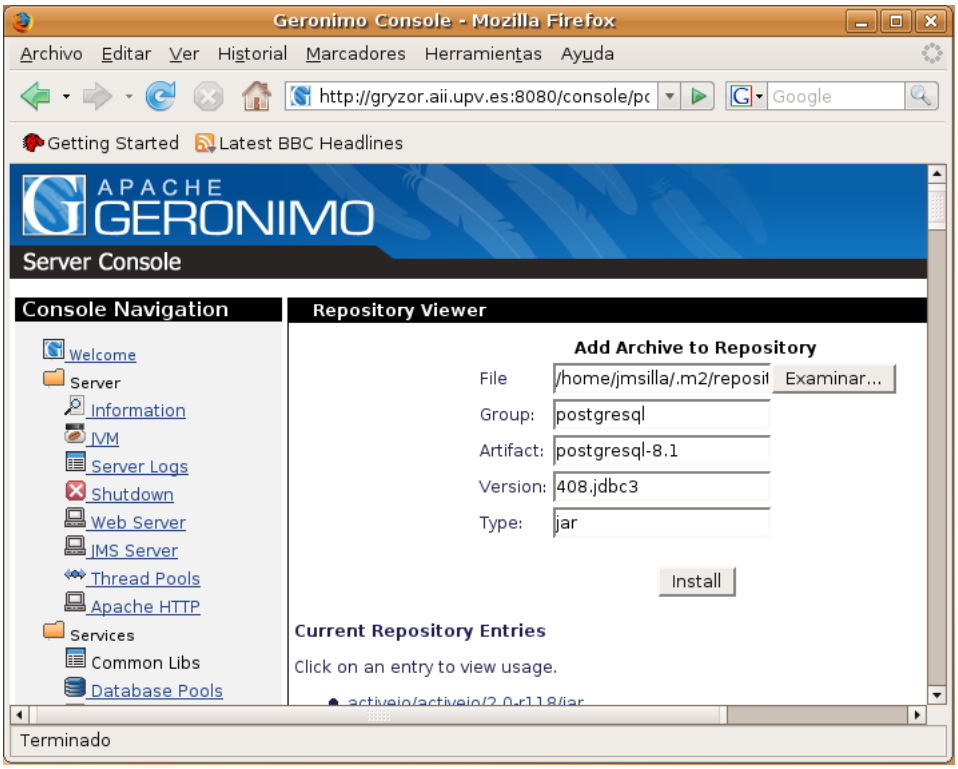

Figura 3.9. Rellenar formulario de librería

Al finalizar este proceso, y si no recibimos ningún mensaje de error, ya tenemos la nueva librería añadida al repositorio de Geronimo. Una vez añadida, debemos modificar el descriptor de Geronimo de nuestro proyecto para indicarle que utilice esa librería. Para ello, en la lista de librerías que aparece al hacer click en Common Libs buscamos la que acabamos de añadir. El nombre del enlace a buscar estará en formato *group/artifact/version/type*, según se haya rellenado el formulario en el paso de subir la librería correspondiente a dicho enlace. Hacemos click en el enlace correspondiente.

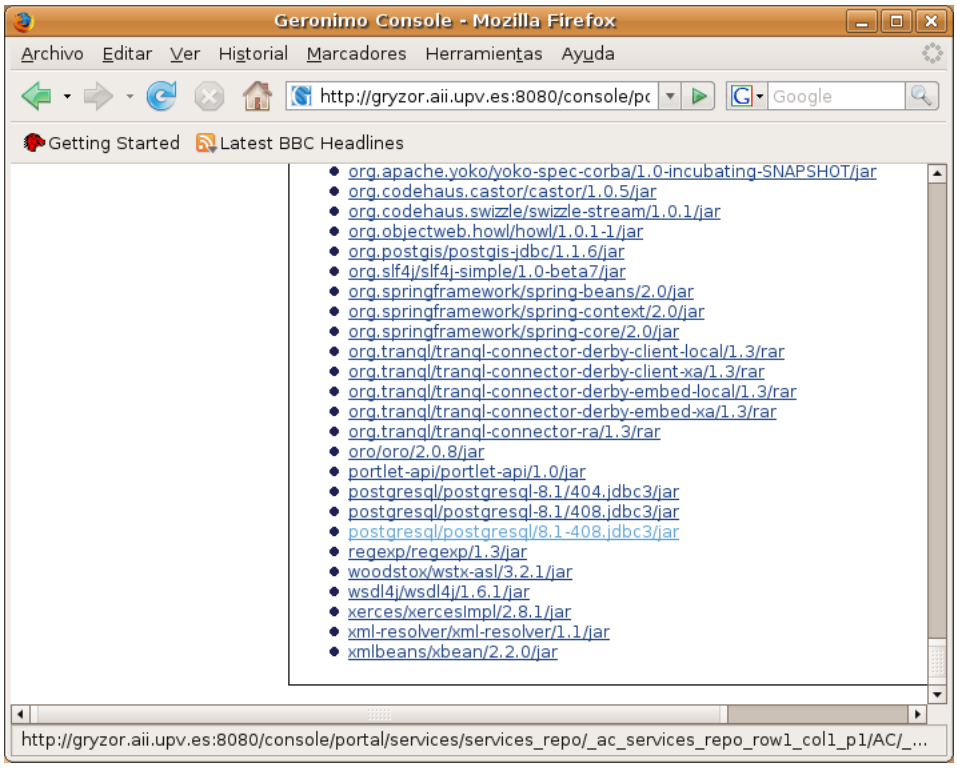

Figura 3.10. Enlaces de las librerías

En el siguiente paso nos aparece un pequeño trozo de código XML. Lo que tenemos que hacer es añadir la sección `<dependency>` que se muestra dentro de la sección `<dependencies>` de nuestro fichero `geronimo-web.xml`. Así ya tendremos añadida la dependencia de nuestro proyecto con esa librería.

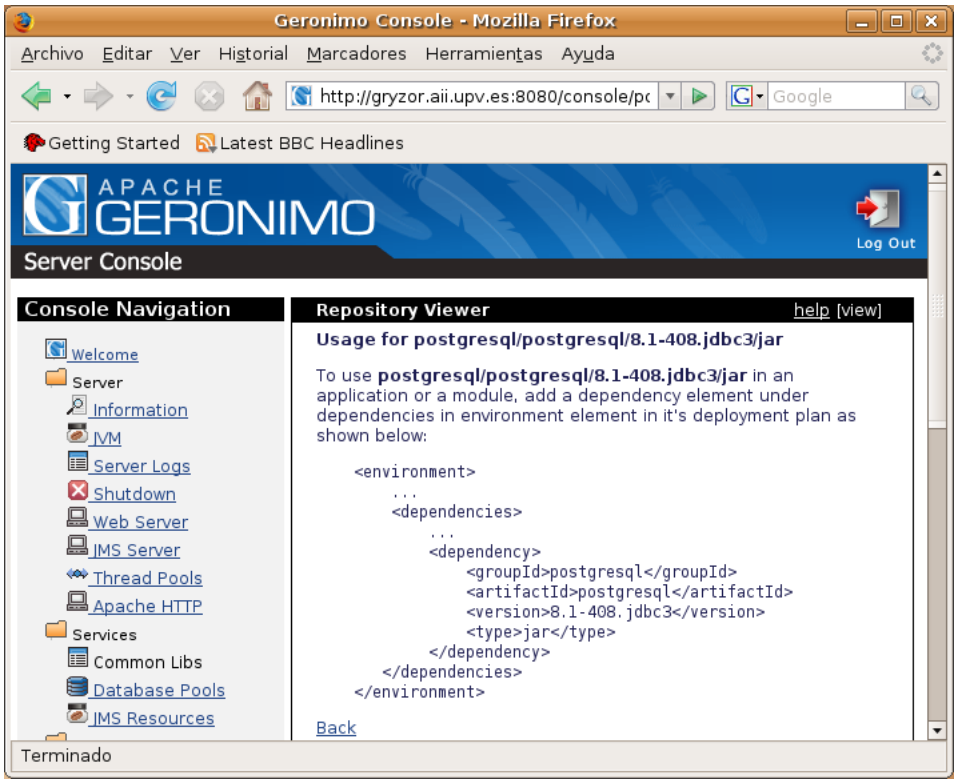

Figura 3.11. XML que añadir al descriptor de Geronimo

En el ejemplo de las imágenes hemos añadido la librería `postgresql-8.1-408.jdbc3.jar`, que es un driver JDBC para PostgreSQL 8.1 más reciente que el que instala Geronimo si lo descargamos durante el proceso de creación de un Database Pool.